

Build COMPETENCY  
across your TEAM



**Microsoft Partner**  
Gold Cloud Platform  
Silver Learning

## Day 4. JQuery



Smita B Kumar



## JQuery- Topics Overview

- Introduction to jQuery
- jQuery selectors
- jQuery Events
- jQuery HTML Manipulation
- jQuery CSS Manipulation
- jQuery DOM Manipulation
- jQuery Effects
- jQuery Animations

# Introduction to jQuery

- jQuery is a lightweight, "write less, do more", JavaScript library.
- The purpose of jQuery is to make it much easier to use JavaScript on your website.
- jQuery takes a lot of common tasks that require many lines of JavaScript code to accomplish, and wraps them into methods that you can call with a single line of code.
- jQuery also simplifies a lot of the complicated things from JavaScript, like AJAX calls and DOM manipulation.
- The jQuery library contains the following features:
  - HTML/DOM manipulation
  - CSS manipulation
  - HTML event methods
  - Effects and animations
  - AJAX
  - Utilities
- **Tip:** In addition, jQuery has plugins for almost any task out there.

# Introduction to jQuery- Why jQuery?

## Why jQuery?

- There are lots of other JavaScript frameworks out there, but jQuery seems to be the most popular, and also the most extendable.
- Many of the biggest companies on the Web use jQuery, such as:
  - Google
  - Microsoft
  - IBM
  - Netflix
- **Will jQuery work in all browsers?**

The jQuery team knows all about cross-browser issues, and they have written this knowledge into the jQuery library. jQuery will run exactly the same in all major browsers.

# Introduction to jQuery- Adding jQuery

## Adding jQuery to Your Web Pages

- There are several ways to start using jQuery on your web site. You can:
  - Download the jQuery library from [jquery.com](http://jquery.com)
  - Include jQuery from a CDN, like Google

```
<head>  
<script src="jquery-3.3.1.min.js"></script>  
</head>
```

## Downloading jQuery

- There are two versions of jQuery available for downloading:
  - Production version - this is for your live website because it has been minified and compressed
  - Development version - this is for testing and development (uncompressed and readable code)
- Both versions can be downloaded from [jQuery.com](http://jquery.com).

## jQuery CDN

- If you don't want to download and host jQuery yourself, you can include it from a CDN (Content Delivery Network).
- Both Google and Microsoft host jQuery.

```
<head>  
<scriptsrc="https://ajax.googleapis.com/ajax/lib  
s/jquery/3.3.1/jquery.min.js"></script>  
</head>
```

## Introduction to jQuery- jQuery Syntax

The jQuery syntax is tailor-made for **selecting** HTML elements and performing some **action** on the element(s).

Basic syntax is: **`$(selector).action()`**

- A \$ sign to define/access jQuery
- A (*selector*) to "query (or find)" HTML elements
- A jQuery *action()* to be performed on the element(s)

Examples:

`$(this).hide()` - hides the current element.

`$("p").hide()` - hides all `<p>` elements.

`$(".test").hide()` - hides all elements with `class="test"`.

`$("#test").hide()` - hides the element with `id="test"`.

# Introduction to jQuery- The Document Ready Event

- This is to prevent any jQuery code from running before the document is finished loading (is ready).
- It is good practice to wait for the document to be fully loaded and ready before working with it. This also allows you to have your JavaScript code before the body of your document, in the head section.
- Here are some examples of actions that can fail if methods are run before the document is fully loaded:

- Trying to hide an element that is not created yet
- Trying to get the size of an image that is not loaded yet

```
$(document).ready(function(){  
    // jQuery methods go here...  
});
```

- **Tip:** The jQuery team has also created an even shorter method for the document ready event:

```
$(function(){  
    // jQuery methods go here...  
});
```

# jQuery Selectors

- jQuery selectors allow you to select and manipulate HTML element(s).
- jQuery selectors are used to "find" (or select) HTML elements based on their name, id, classes, types, attributes, values of attributes and much more. It's based on the existing [CSS Selectors](#), and in addition, it has some own custom selectors.
- All selectors in jQuery start with the dollar sign and parentheses: \$().

## The element Selector

- The jQuery element selector selects elements based on the element name.
- You can select all <p> elements on a page like this:

```
<!DOCTYPE html>
<html><head><script
src="https://ajax.googleapis.com/ajax/libs/jq
uery/3.3.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("p").hide();
    });
});
</script></head><body>
<h2>This is a heading</h2>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
<button>Click me to hide
paragraphs</button>
</body>
</html>
```



## The #id Selector

- The jQuery #id selector uses the id attribute of an HTML tag to find the specific element.
- An id should be unique within a page, so you should use the #id selector when you want to find a single, unique element.
- To find an element with a specific id, write a hash character, followed by the id of the HTML element:

`$("#test")`

```
<!DOCTYPE html>
<html><head><script
src="https://ajax.googleapis.com/ajax/libs/jq
uery/3.3.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("#test").hide();
    });
});
</script>
</head>
<body>
<h2>This is a heading</h2>
<p>This is a paragraph.</p>
<p id="test">This is another paragraph.</p>
<button>Click me</button>
</body>
</html>
```

# jQuery Selectors-The .class Selector

## The .class Selector

- The jQuery class selector finds elements with a specific class.
- To find elements with a specific class, write a period character, followed by the name of the class: `$(".test")`

```
<!DOCTYPE html>
<html><head><script
src="https://ajax.googleapis.com/ajax/libs/jq
uery/3.3.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $(".test").hide();
    });
});
</script>
</head>
<body>
<h2 class="test">This is a heading</h2>
<p class="test">This is a paragraph.</p>
<p>This is another paragraph.</p>
<button>Click me</button>
</body>
</html>
```

# jQuery Selectors-Selector Lists

| Syntax                                 | Description  |
|--|--|
| <code>\$("*")</code>                   | Selects all elements   |
| <code>\$(this)</code>                  | Selects the current HTML element   |
| <code>\$("p.intro")</code>             | Selects all <code>&lt;p&gt;</code> elements with <code>class="intro"</code>  |
| <code>\$("p:first")</code>             | Selects the first <code>&lt;p&gt;</code> element   |
| <code>\$("ul li:first")</code>         | Selects the first <code>&lt;li&gt;</code> element of the first <code>&lt;ul&gt;</code>                                 |
| <code>\$("ul li:first-child")</code>   | Selects the first <code>&lt;li&gt;</code> element of every <code>&lt;ul&gt;</code>                                     |
| <code>\$("[href]")</code>              | Selects all elements with an href attribute  |
| <code>\$("a[target='_blank']")</code>  | Selects all <code>&lt;a&gt;</code> elements with a target attribute value equal to <code>"_blank"</code>               |
| <code>\$("a[target!='_blank']")</code> | Selects all <code>&lt;a&gt;</code> elements with a target attribute value NOT equal to <code>"_blank"</code>           |
| <code>\$(":button")</code>             | Selects all <code>&lt;button&gt;</code> elements and <code>&lt;input&gt;</code> elements of <code>type="button"</code> |
| <code>\$("tr:even")</code>             | Selects all even <code>&lt;tr&gt;</code> elements  |
| <code>\$("tr:odd")</code>              | Selects all odd <code>&lt;tr&gt;</code> elements   |

# jQuery Events

- What are Events?
- All the different visitor's actions that a web page can respond to are called events.
- An event represents the precise moment when something happens.
- Examples:
  - moving a mouse over an element
  - selecting a radio button
  - clicking on an element
- The term "**fires/fired**" is often used with events. Example: "The keypress event is fired, the moment you press a key".

# jQuery Events-Common DOM events

| Mouse Events | Keyboard Events | Form Events | Document/Window Events |
|--------------|-----------------|-------------|------------------------|
| click        | keypress        | submit      | load                   |
| dblclick     | keydown         | change      | resize                 |
| mouseenter   | keyup           | focus       | scroll                 |
| mouseleave   |                 | blur        | unload                 |

## jQuery Syntax For Event Methods

In jQuery, most DOM events have an equivalent jQuery method. To assign a click event to all paragraphs on a page, you can do this:

```
$("#p").click(function(){  
    // action goes here!!  
});
```

# jQuery Events-Commonly Used jQuery Event Methods

## **\$(document).ready()**

The `$(document).ready()` method allows us to execute a function when the document is fully loaded.

## **click()**

The `click()` method attaches an event handler function to an HTML element.

## **dblclick()**

The `dblclick()` method attaches an event handler function to an HTML element.

## **mouseenter()**

The `mouseenter()` method attaches an event handler function to an HTML element.

## **mouseleave()**

The `mouseleave()` method attaches an event handler function to an HTML element.

## **mousedown()**

The `mousedown()` method attaches an event handler function to an HTML element.

## **mouseup()**

The `mouseup()` method attaches an event handler function to an HTML element.

## **hover()**

The `hover()` method takes two functions and is a combination of the `mouseenter()` and `mouseleave()` methods.

## **focus()**

The `focus()` method attaches an event handler function to an HTML form field.

## **blur()**

The `blur()` method attaches an event handler function to an HTML form field.

# jQuery Events-Commonly Used jQuery Event Methods

```
<!DOCTYPE html>
<html><head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("p").on({
        mouseenter: function(){
            $(this).css("background-color", "lightgray");
        },
        mouseleave: function(){
            $(this).css("background-color", "lightblue");
        },
        click: function(){
            $(this).css("background-color", "yellow");
        }
    });
});
</script>
</head><body>
<p>Click or move the mouse pointer over this paragraph.</p>
</body></html>
```

# jQuery HTML Manipulation

- jQuery contains powerful methods for changing and manipulating HTML elements and attributes.
- **jQuery DOM Manipulation**
  - One very important part of jQuery is the possibility to manipulate the DOM.
  - jQuery comes with a bunch of DOM related methods that make it easy to access and manipulate elements and attributes.
- **Get Content - text(), html(), and val()**

Three simple, but useful, jQuery methods for DOM manipulation are:

  - text() - Sets or returns the text content of selected elements
  - html() - Sets or returns the content of selected elements (including HTML markup)
  - val() - Sets or returns the value of form fields

```
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("#btn1").click(function(){
        alert("Text: " + $("#test").text());
    });
    $("#btn2").click(function(){
        alert("HTML: " + $("#test").html());
    });
    $("button").click(function(){
        alert("Value: " + $("#test").val());
    });
    $("button").click(function(){
        alert($("#s1").attr("href"));
    });
});
</script></head><body>
<p id="test">This is some <b>bold</b> text in a paragraph.</p>
<button id="btn1">Show Text</button>
<button id="btn2">Show HTML</button>
<p>Name: <input type="text" id="test" value="Mickey Mouse"></p>
<button id="btn3">Show Value</button>
<p><a href="https://www.synergetics-live.com" id="s1">Synergetics-India
Live</a></p>
<button id="btn4">Show href Value</button>
</body>
```



# jQuery - Add Elements

## Add New HTML Content

We will look at four jQuery methods that are used to add new content:

- `append()` - Inserts content at the end of the selected elements
  - `prepend()` - Inserts content at the beginning of the selected elements
  - `after()` - Inserts content after the selected elements
  - `before()` - Inserts content before the selected elements
- `$("p").append("Some appended text.");`
  - `$("p").prepend("Some prepended text.");`
  - `$("img").after("Some text after");`
  - `$("img").before("Some text before");`

```
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js
"></script>
<script>
function afterText() {
    var txt1 = "<b>I </b>";           // Create element with HTML
    var txt2 = $("<i></i>").text("love "); // Create with jQuery
    var txt3 = document.createElement("b"); // Create with DOM
    txt3.innerHTML = "jQuery!";
    $("img").after(txt1, txt2, txt3);    // Insert new elements after img
}
</script>
</head>
<body>



<p>Click the button to insert text after the image.</p>

<button onclick="afterText()">Insert after</button>
```

# jQuery - Remove Elements

## Remove Elements/Content

To remove elements and content, there are mainly two jQuery methods:

- `remove()` - Removes the selected element (and its child elements)
- `empty()` - Removes the child elements from the selected element
  - `$("#div1").remove();`
  - `$("#div1").empty();`

# jQuery CSS Manipulation

jQuery has several methods for CSS manipulation. We will look at the following methods:

- jQuery **addClass()** Method- Adds one or more classes to the selected elements
  - ```
$("#button").click(function(){  
    $("h1, h2, p").addClass("blue");  
    $("div").addClass("important");  
});
```
- jQuery **removeClass()**- Removes one or more classes from the selected elements
  - ```
$("#button").click(function(){  
    $("h1, h2, p").removeClass("blue");  
});
```

# jQuery CSS Manipulation

jQuery has several methods for CSS manipulation. We will look at the following methods:

- jQuery **toggleClass()** Method -Toggles between adding/removing classes from the selected elements
  - `$("button").click(function(){  
 $("h1, h2, p").toggleClass("blue");  
});`
- jQuery **css()** Method - Sets or returns the style attribute
  - `$("p").css("background-color");`
  - `$("p").css({"background-color": "yellow", "font-size": "200%"});`

# jQuery Effects- Hide, Show, Toggle, Slide, Fade, and Animate.

## jQuery Effects - Hide, Show, Toggle, Slide, Fade, and Animate.

- jQuery hide() and show()

- ✓ With jQuery, you can hide and show HTML elements with the hide() and show() methods:

- ✓ Syntax: `$(selector).hide(speed,callback);`

- `$("#p").hide(1000);`

- `$("#p").show(1000);`

- jQuery toggle()

- With jQuery, you can toggle between the hide() and show() methods with the toggle() method.

- Shown elements are hidden and hidden elements are shown

- Syntax: `$(selector).toggle(speed,callback);`

# jQuery Fading Effects

With jQuery you can fade an element in and out of visibility.

jQuery has the following fade methods:

- fadeIn()
  - fadeOut()
  - fadeToggle()
  - fadeTo()
- **Syntax:**
    - `$(selector).fadeIn(speed,callback);`
  - **Syntax:**
    - `$(selector).fadeTo(speed,opacity,callback);`



# jQuery Effects - Sliding

## jQuery Sliding Methods

With jQuery you can create a sliding effect on elements.

jQuery has the following slide methods:

- `slideDown()`
- `slideUp()`
- `slideToggle()`
- **Syntax:**
  - `$(selector).slideDown(speed,callback);`



# jQuery Animations

- The jQuery animate() method is used to create custom animations.
- **Syntax:**
  - `$(selector).animate({params},speed,callback);`
- The required params parameter defines the CSS properties to be animated.
- The optional speed parameter specifies the duration of the effect. It can take the following values: "slow", "fast", or milliseconds.
- The optional callback parameter is a function to be executed after the animation completes.



Q & A

Contact: [smitakumar@synergetics-india.com](mailto:smitakumar@synergetics-india.com)



# Thank You



Contact: [smitakumar@synergetics-india.com](mailto:smitakumar@synergetics-india.com)